

条件分岐、繰り返し、 例外処理

初めてのプログラミング

2019年度

只木進一（理工学部）

サンプルプログラムの取得

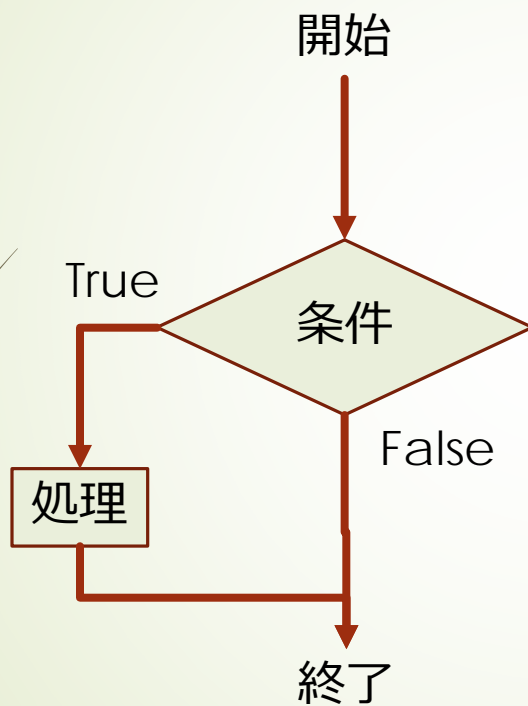
- GitHubRepositoryを指定
 - <https://github.com/first-programming-saga/Control>

- ➡ ここまでの例は、開始から終了まで一直線
- ➡ 実際のプログラムでは
 - ➡ 条件にあうところだけ実行
 - ➡ 操作を繰り返す
 - ➡ 異常なことが起こった時の対処

インデント

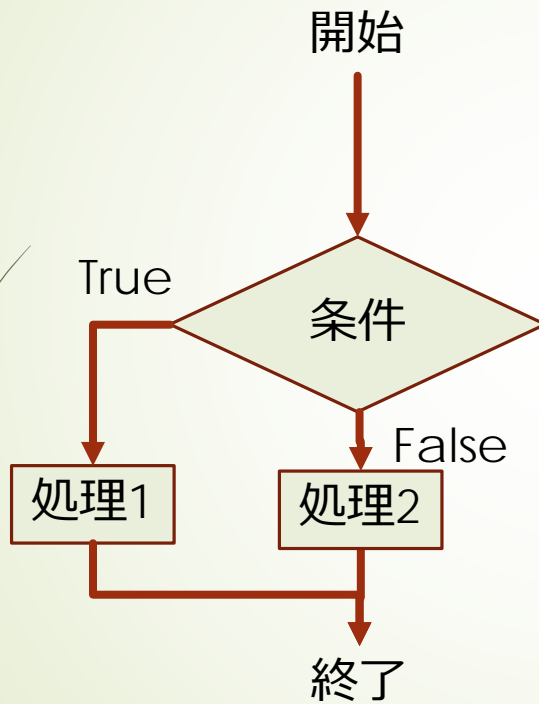
- ▶ Pythonでは、インデントを使って、プログラムブロックを区別
 - ▶ 条件を満たした場合に実行する部分
 - ▶ 繰り返し部分

条件分岐1



```
a = -1  
if a < 0: #a が負  
    a = 0  
print(a)
```

条件分岐2



```
a = -1
if a < 0 : #a が負
    a = 0
else :
    a = 1
print(a)
```

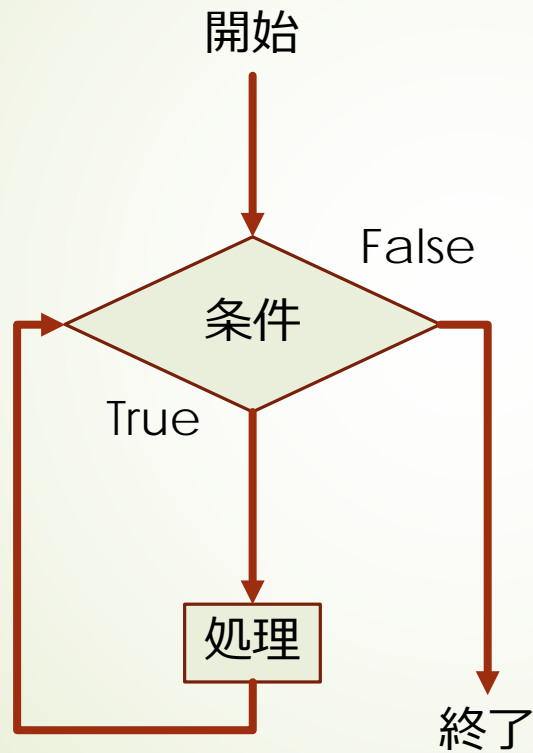
条件分岐

- ▶ ifを入れ子にする方法とelifを使う方法
 - ▶ 自分にとってわかりやすいように

Control/if.ipynb

while

条件を満たす間繰り返す



```
a = -1
while a < 10 : #aが10未満
    a += 1
    print(a)
```


while

条件を満たす間繰り返す

```
from random import randint
x = randint(0,100) # 0以上100以下の整数乱数
while x < 80:
    print(x)
    x = randint(0,100)
print("end")
```

➡ 無限ループに注意

breakとcontinue

- break : ループを終了
- continue : ループを中断して次の繰り返しへ

Control/while.ipynb

```
from random import randint
while True:
    x = randint(0,99)
    if x % 5 == 0:
        break
    elif x % 7 == 0:
        continue
    print (x)
print("end with "+str(x))
```

while-else

```
m=int(math.sqrt(n))
k = 3
while k <= m:
    if n % k == 0:
        print(f'{n}は{k}で割り切れるため、素数ではない')
        break # whileループから抜け出す
    k += 2
else: # whileループの最後まで至った場合
    print(f'{n}は素数である')
```

終了

Control/while.ipynb

for 範囲内を繰り返す

➡ 指定回数繰り返す

```
for x in range(n) : #n回繰り返す  
    処理
```

➡ range (開始、終了、ステップ)

```
for x in range(0,n,2) : #0からn-1まで、一つおきに繰り返す  
    処理
```

```
#0から10までの和  
sum = 0  
for x in range(0,11)  
    sum += x  
print(sum)
```

Control/for0.ipynb

for 範囲内を繰り返す

- ▶ リストなどの要素で繰り返し範囲を指定

```
colors = ['red', 'green', 'blue']  
for c in colors:  
    print(colors)
```

例外処理

- 実行中にエラーが発生すると、プログラムの実行は、そこで停止。
- 予期しないデータ、ディスクの状態、ネットワークの状態で、エラーが発生する可能性
- 予め、エラー発生を予測し、エラー発生時の対応を記述しておくことで、その後の処理を継続

例外処理

```
try:  
    エラー発生可能性のある処理  
except:  
    エラー時の処理
```

➡ エラーの種類毎に振り分け

```
try:  
    エラー発生可能性のある処理  
except 例外1:  
    例外1時の処理  
except 例外2:  
    例外2時の処理
```

次回

➡ 6章「リスト」